

Système de Gestion de Base de Données (SGBD)

Les éléments de modélisation Objet-Relationnel

Introduction

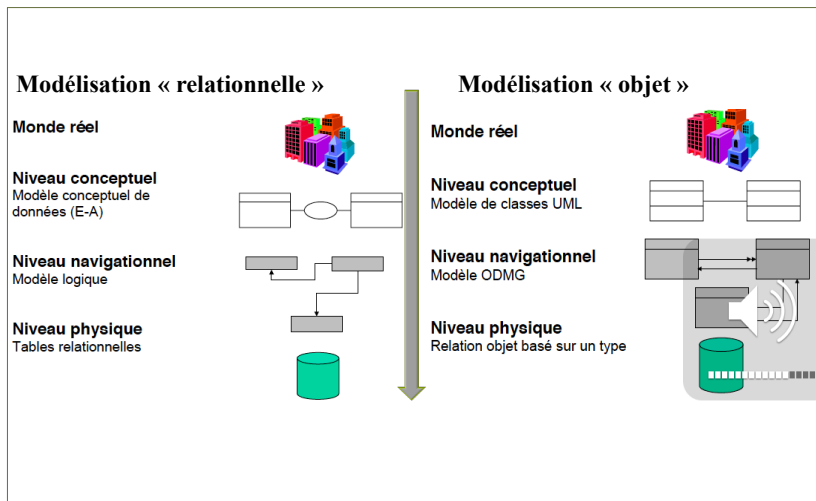
Nouvelles possibilités avec le modèle objet-relationnel :

- définition de nouveaux types complexes avec fonctions de manipulation
- une colonne peut contenir une collection : liste, ensemble, . . .
- ligne considérée comme un objet avec un identificateur OID
- utilisation de références aux objets
- extension de SQL : SQL3 (ou SQL99) recherche et manipulation des données

Inconvénients du modèle objet-relationnel :

- ne s'appuie pas sur une théorie solide comme le modèle relationnel
- pas de standardisation : implantations différentes, partielles dans les SGBD

Introduction



Le modèle navigationnel

Objectifs

- Faciliter le passage vers l'Objet par un ensemble de règles de transposition : Diagramme de classe-UML => navigationnel => Objet
- Favoriser la visualisation des structures complexes et des chemins de navigation possibles en conservant la notion d'objet.
- Représenter concrètement le plus de contraintes possibles pouvant être spécifiées par le langage DDL du SGBD.
- **Lacune du Mnav :**
 - Ne représente pas bien le partage des types entre différentes parties du modèle UML (classes).
 - Devient rapidement encombré et encombrant pour des modèles complexes!

Pourquoi le navigationnel? Le modèle introduit les liens logiques de type REF (implémentation de la REF est toujours indéterminée au niveau conceptuel).

Les références (Ref ou REF) facilitent la navigation entre les objets). Les contraintes au niveau de la référence peut être spécifiée dans le modèle.

Le modèle navigationnel

- La classe UML sera rendue dans le Mnav par **une classe** dite **externe** qui est en quelque sorte au plan informatique un type (i.e. une structure nommée) défini par ses attributs.
- Une **classe externe** définit des objets autonomes (dont l'existence ne dépend pas d'un autre objet) dans la base i.e. possède une valeur + un oid
- Un **oid** est un identifiant d'objet interne et unique, non réutilisable pour toute la vie de la base. Un objet ayant un oid est persistant dans la base.
- Les **attributs** du modèle Mnav sont simples ou complexes.

Attribut simple:

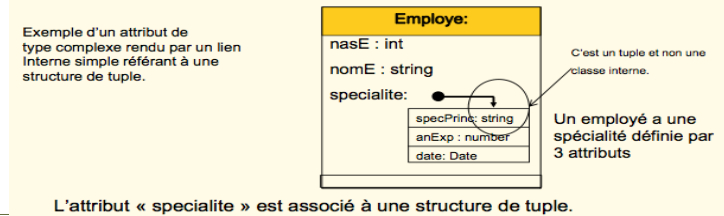
L'attribut simple est idem à celui de UML: type atomique avec ses opérations primitives régulières et implicites:

Exemple: un entier avec ses opérateurs arithmétiques, une chaîne avec son opérateur de concaténation, ...

Le modèle navigationnel

Attribut complexe:

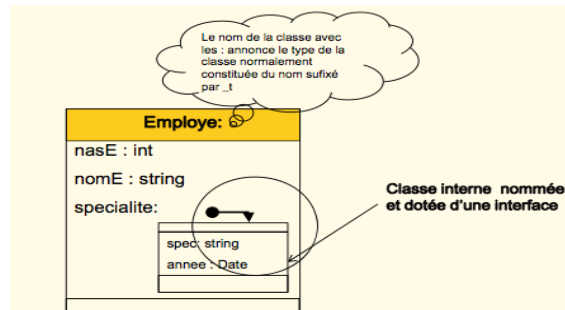
- L'attribut complexe permet d'abandonner la FN1 en ajoutant une structure de tuple pour regrouper plusieurs attributs.
- Le tuple est une structure logique linéaire formée d'attributs hétérogènes ou pas, simples ou complexes, sans interface.
- Un attribut complexe définit des valeurs dont l'existence dépend que de l'objet externe qui l'héberge. Il est représenté par une structure non nommée et sans interface. Ce n'est pas une classe interne mais uniquement une structure pour y ranger des valeurs.



Le modèle navigationnel

Attribut associé à un seul objet:

- Un attribut associé à un seul objet d'une classe est représenté par un lien interne avec un lien monovalué vers une classe interne.

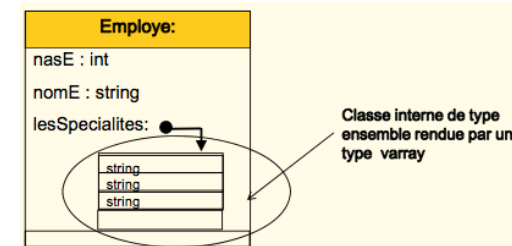


Dans cet exemple, l'attribut specialite est obligatoirement valué par un objet qui caractérise chaque Employee et est imbriqué dans la classe externe Employee. Cet objet a une interface et une persistance.

Le modèle navigationnel

Attribut d'ensemble ordonné:

- Un attribut **d'ensemble ordonné** est rendu par un lien interne **simple** vers une structure de type varray.

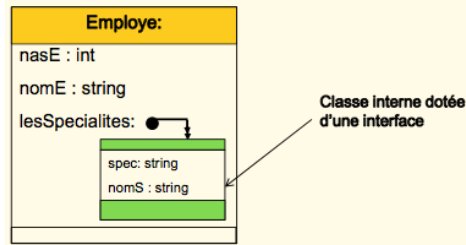


Les éléments de l'attribut lesSpecialites sont stockés dans un varray (similaire à un tableau ou un array). Chaque entrée est indexée et peut-être soit une valeur soit un objet interne. Le varray a des procédures propres dites attributs.

Le modèle navigationnel

Attribut d'ensemble non ordonné:

- Un attribut **d'ensemble** est rendu par un lien interne **multiple** vers un ensemble d'objets d'une classe interne.



Les éléments de l'attribut *lesSpecialites* sont des objets de la classe interne pouvant être simples ou complexes imbriqués (ou pas) dans la classe externe *Employe*.

Du DC-UML vers le navigationnel: règles de passage

Transformation d'une classe UML :

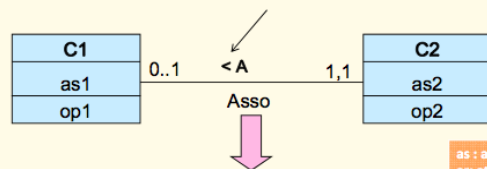
- Chaque classe du DC-UML, sauf les classes-associations, est transformée en une classe externe du Mnav.
- Chaque attribut composé de type tuple UML devient une classe interne avec un lien interne simple (une seule occurrence)
- Chaque attribut d'ensemble est rendu par un lien **interne multiple** ou un lien **externe multiple** (avec Ref) selon l'existence autonome ou pas des objets au regard de la suppression. Le choix est aussi fonction de la performance d'accès recherchée.

Identifiant facultatif pour la classe du Mnav:

- Chaque classe du Mnav (sauf les sous-classes dans l'arbre d'héritage) et la classe d'association, peuvent avoir un identifiant de valeur. Souvent pratique pour la recherche, mais non obligatoire.
- Si aucun attribut ne convient pas comme clé, on peut ajouter une clé, si possible significative. Ce n'est cependant pas une exigence du modèle.

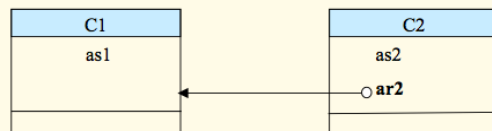
Du diagramme de classes vers le modèle navigationnel

Transformation des associations 0..1-1..1:



1ère solution :

L'attribut de type Ref peut être nul dans C2. **L'accès privilégié est par C2.**



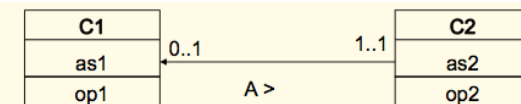
Toutes les multiplicités de la classe UML ne sont pas représentées par ce Mnav!

En lisant l'attribut *ar2* de C2 (1 ref) il y a un accès direct à l'instance correspondante dans C1. Le *ar2* est de type REF et peut être nul.

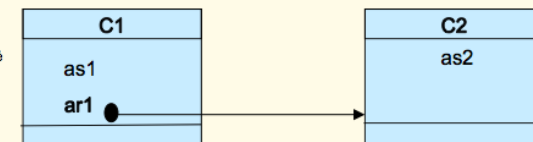
Du diagramme de classes vers le modèle navigationnel

Transformation des associations 0..1-1..1: (Suite)

2ème solution avec le parcours inverse : accès via C1 et accès à C2 via l'attribut de référence *ar1* qui ne peut pas être nul.



La multiplicité du côté de C1 n'est pas représentée !



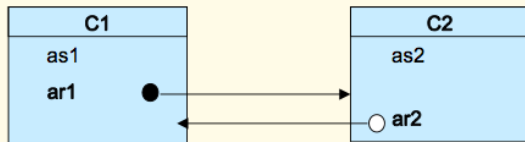
Le sens de la navigation est indiqué dans le DC-UML suggérant la transformation appropriée.

Si le sens de la navigation est absent dans l'association : on peut assumer alors que les 2 sens sont équi probables

Du diagramme de classes vers le modèle navigationnel

Transformation des associations 0..1-1..1: (Suite)

3e solution pour les 2 sens de parcours: 2 liens simples (aucun sens privilégié)



ar1 de C1 est du type REF monovalué non null qui réfère à C2 (donc aux objets de la classe C2).

ar2 : est un attribut REF qui réfère à un objet de C1 et qui peut être null.

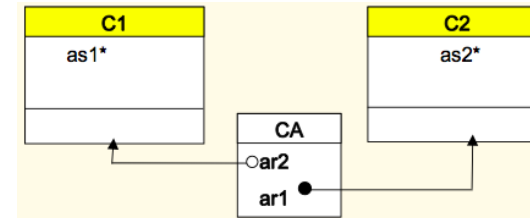
- La suppression de C2 et 1 mise à jour de ar1.
- La suppression de C1 est suivie d'une mise à jour de ar2

Ce Mnav exprime le renforcement des 2 contraintes par le modèle sans égard au sens privilégié du parcours de l'association.

Du diagramme de classes vers le modèle navigationnel

Transformation des associations 0..1-1..1: (Suite)

4e solution (dite universelle): utilisation d'une 3e classe CA avec 2 liens externes. Toutes les multiplicités sont renforcées par le modèle.



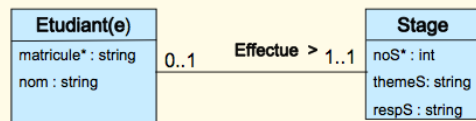
Evolution du modèle:

Le modèle peut évoluer en intégrant un attribut pour la classe d'association sans pénaliser l'interrogation et la manipulation de la base d'objets.

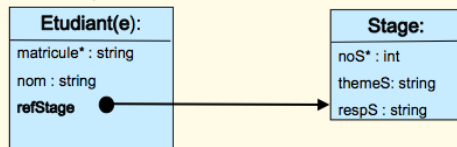
Du diagramme de classes vers le modèle navigationnel

Transformation des associations 0..1-1..1: Exemple

Stages étudiants (UML) :



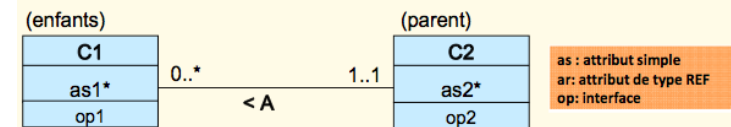
Solution avec une ref simple:



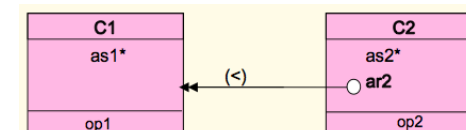
Un lien peut être parcouru par la référence : avec la notation pointée.

Du diagramme de classes vers le modèle navigationnel

Association de multiplicités 0..* et 1..1



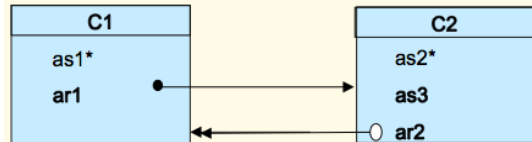
1ère solution : un lien multiple. La multiplicité minimale 0 figure à la source du lien ar2 de C2. Le parcours privilégié est pris en compte.



Du diagramme de classes vers le modèle navigationnel

Association de multiplicités 0..* et 1..1

2e solution : un lien multiple et un lien simple externe inverse.



ar2 lien multiple donc ensemble de références sur C1 (éventuellement vide).

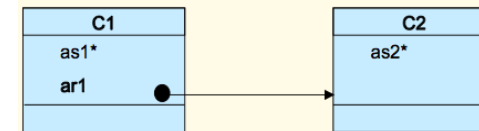
ar1 est un attribut de type Ref qui référence un objet de C2. (toujours valué)

- La suppression d'un objet de C2 doit entraîner les objets de C1 associés.
- La suppression d'un objet C1? C1 disparaît et sa ref supprimée via le lien *ar2*.

Du diagramme de classes vers le modèle navigationnel

Association de multiplicités 0..* et 1..1

3e solution : un lien monovalué non nul vers le parent C2

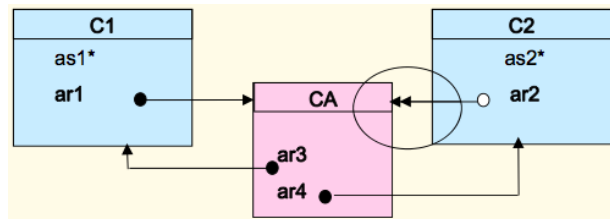


- Chaque instance de C1 est liée à une instance de C2 par la référence *ar1* obligatoire.
- La suppression de C1 se limite à cette opération.
- La suppression de C2 doit être bloquée si elle est référée par un objet de C1. La source du lien *ar1* doit avoir obligatoirement une référence valide.
- La suppression de C2 devrait donc éventuellement entraîner aussi celle de C1 qui s'y réfère et cela au cours d'une même transaction.

Du diagramme de classes vers le modèle navigationnel

Association de multiplicités 0..* et 1..1

4e solution : (universelle, sans parcours privilégié): utilisation d'une troisième classe. Cette classe porte le nom de l'association et contient deux liens simples pour implémenter l'association.



Sans les liens supplémentaires vers CA, la navigation (donc l'accessibilité) ne serait possible qu'à partir de CA. Cette solution fournit un Mnav plus lourd à gérer! La suppression de C2 peut impliquer plusieurs changements: suppression dans CA et dans C1.

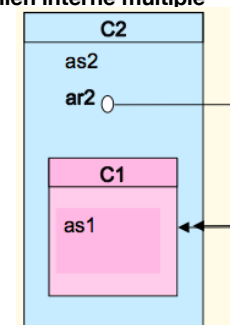
Du diagramme de classes vers le modèle navigationnel

Association de multiplicités 0..* et 1..1 avec un lien interne multiple

Autre modèle légèrement différent.

En supprimant un objet C2, les objets C1 associés sont aussi supprimés.

L'ajout d'un objet de C1 suppose l'existence d'un objet hôte C2.

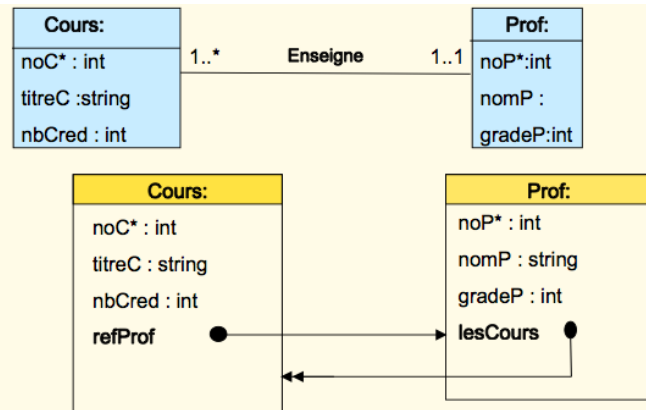


Le lien interne et multiple pour *ar2* sous-tend un ensemble formé avec des objets- colonnes de type C1 imbriqués. La réalisation physique en Oracle prend la forme d'une sous-table d'objets.

L'accès est privilégié par C2 pour trouver les C1. La suppression de C1 n'entraîne pas celle de C2, la classe hôte.

Du diagramme de classes vers le modèle navigationnel

Association de multiplicités 0..* et 1..1: Exemple



Quelles sont les contraintes du Mnav pour l'ajout et la suppression d'un cours?

Du diagramme de classes vers le modèle navigationnel

Association de multiplicités 0..* et 1..1: Exemple

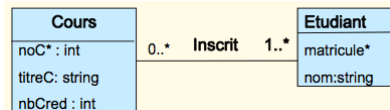
Quelles sont les contraintes associées à l'ajout et à la suppression d'un cours / prof?

- À l'ajout d'un cours:
Le cours ajouté doit être associé à un prof par une référence obligatoire.
- À la suppression d'un cours:
Le cours est supprimé.
Il doit être aussi supprimé de la liste des refs du prof concerné.
- À l'ajout d'un professeur:
Cela implique l'ajout d'un ou plusieurs cours (contrainte de multiplicité).
- À la suppression d'un professeur:
- suppression des cours enseignés par ce prof.
et - suppression du prof.

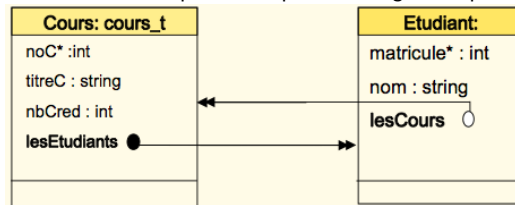
Ces contraintes devront être implémentées dans une méthode de suppression qui prendra en charge la vérification de ces contraintes dans une même transaction.

Du diagramme de classes vers le modèle navigationnel

Association 0..* et 1..* sans attributs d'association



Sans préférence de parcours: deux liens multiples avec deux nouveaux attributs de type REF. Une redondance de liens qui rend cependant la gestion plus lourde!



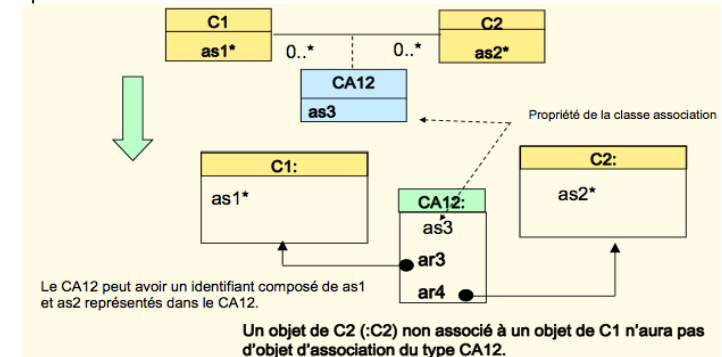
Suppression d'un étudiant : danger d'incohérence avec l'oubli de la mise à jour de l'attribut lesEtudiants. L'application doit prendre à sa charge cette dernière mise à jour.

- En suppression d'un cours ? Idem

Du diagramme de classes vers le modèle navigationnel

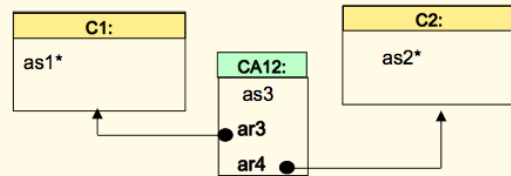
Association 0..* et 0..* avec attributs à la classe-association

1ère solution: (universelle) : une 3e classe avec le nom de l'association est formée avec les attributs de l'association, deux liens monovalués vers les classes participantes.



Du diagramme de classes vers le modèle navigationnel

Association 0..* et 0..* avec attributs à la classe-association



Quels sont les C1 associés à une valeur clé k2 de C2? – ajoutez la valeur de as3

1- recherche dans CA12
 Select o.ar3.*, o.as3
 From CA12 o
 Where o.ar4.as2 = k2

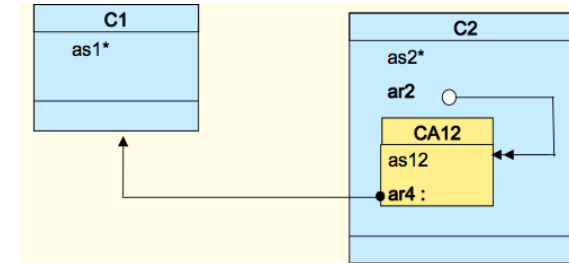
Quels sont les C1 non associés à un objet C2?

2- Les C1 moins les CA12:
 Select x.* From C1
 -- (moins)
 Select o.ar3.*
 From CA12 o
 Where o.ar3 is not null;

Du diagramme de classes vers le modèle navigationnel

Association 0..* et 0..* avec attributs à la classe-association

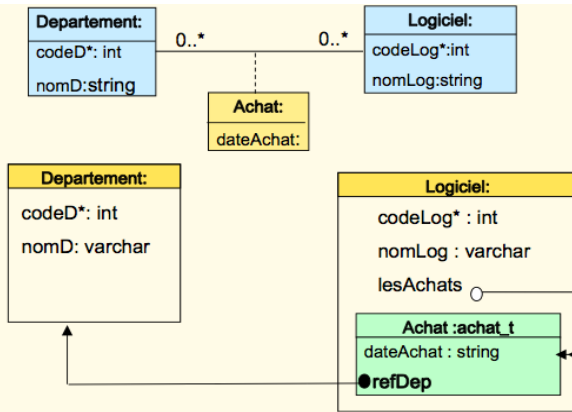
2e solution: une classe interne avec un lien interne multiple et un lien externe simple.



La suppression d'une instance de C1 implique la suppression de l'instance de CA12 associée.

Du diagramme de classes vers le modèle navigationnel

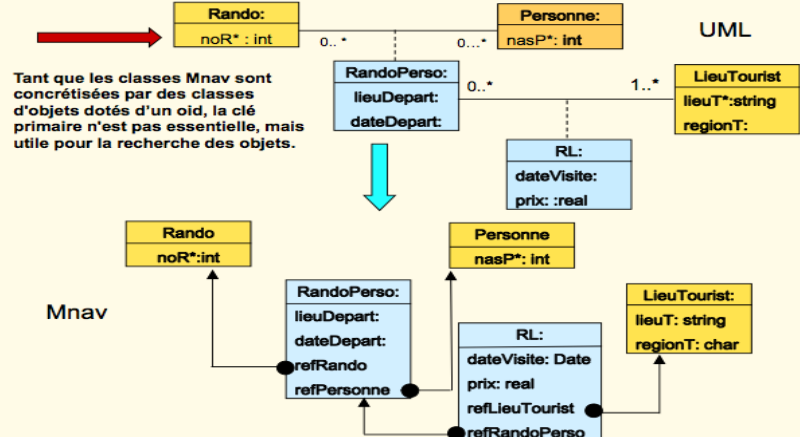
Association 0..* et 0..* avec attributs à la classe-association: Exemple



Les achats de logiciels distincts faits par un département sont représentés par la classe interne dont les éléments ont la structure achat_t

Du diagramme de classes vers le modèle navigationnel

Exemple: Modélisation des randonnées effectuées par des personnes qui visitent des lieux touristiques situés dans diverses régions. (solution universelle)



Du diagramme de classes vers le modèle navigationnel

Exemple: (suite)

Dans le choix de la transformation, il ne faut pas perdre de vue l'exploitation subséquente du modèle. Cela peut justifier la classe d'association.

Voici quelques clauses SQL pour interroger cette base:

- Quels sont les régions de départ pour les randos effectués par le randonneur dont le nasP est 3456?

```
SELECT rp.refRando.noR
FROM RandoPerso rp
WHERE rp.refPersonne.nasP = 3456;
```

- Quels sont les lieux touristiques visités par le randonneur dont le nasP est 3456?

```
SELECT t.refLieuTourist.lieuT
FROM RL t
WHERE t.refRandoPerso.refPersonne.nasP = 3456;
-- double indirection
```

Du diagramme de classes vers le modèle navigationnel

Les choix à privilégier pour les associations

Pour l'association	Modèle navigationnel
1..1 – 1..1	Un lien monovalué obligatoire
1..1 – 0..* (sans préférence de navigation)	Un lien monovalué de enfant vers le parent
0..* - 1..* <u>sans</u> attributs	2 liens multiples
0..* - 1..* <u>avec</u> attributs	2 liens mono + CA

- Les contraintes non exprimées par le Mnav seront renforcées par les méthodes ou par les clauses DDL propres au DDL objet.